This listing of claims will replace all prior versions, and listings, of claim in the application;

IN THE CLAIMS

1. (currently amended) A method, comprising:
inserting an on-processor register allocation instruction within a machine code description of a routine if a function call instruction to perform a function call to another routine is found within said routine, said inserted on processor register allocation instruction to allocate less on-processor register space for the use of said routine than an amount of on-processor register space allocated for the use of said routine by another on-processor register allocation instruction that is executed prior to said inserted on processor register allocation instruction.

2. (original) The method of claim 1 further comprising configuring said allocation instruction to allocate only for the live information that exists within said routine when said inserted allocation instruction is executed.

3. (original) The method of claim 2 wherein said live information is determined by identifying information that is referred to before and after said function call.

4. (original) The method of claim 3 wherein said information identified after said function call extends to an exit block of said routine.

5. (original) The method of claim 4 wherein the worst case path to said exit block is allocated for.

6. (original) The method of claim 3 wherein said information identified after said function call extends to a post-dominator block of said routine.

7. (original) The method of claim 6 wherein the worst case path to said post-dominator block is allocated for.

8. (original) The method of claim 2 wherein said live information is information that is local to said routine.

9. (previously presented) The method of claim 8 wherein a processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and register space used only for global information, said allocation instruction pertaining only to said register space used for local information.

10. (original) The method of claim 2 wherein said live information includes global information.

11. (previously presented) The method of claim 1 wherein said allocation instruction is inserted just before a machine code representation of said function call.

12. (original) The method of claim 1 wherein said allocation instruction is inserted in a pre-dominator basic block of said function call.

13. (original) The method of claim 12 wherein said allocation instruction is inserted in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

14. (currently amended) A method comprising:
inserting an on-processor register allocation instruction within a machine code description of a routine because a functional characteristic selected from the group consisting of:

a) a loop that exists within a control flow graph of said routine;

b) a software pipelined loop; and,

c) a function call to another routine;

is discovered within said routine, said inserted on-processor register allocation instruction to allocate less on processor register space for the use of said routine than had been previously allocated for the use of said routine.

15. (previously presented) The method of claim 14 wherein at least one of said discovered functional characteristics includes said loop that exists within a control flow graph of said routine.

16. (previously presented) The method of claim 15 wherein the allocation instruction inserted for said loop is inserted above said loop in said control flow graph.

17. (original) The method of claim 16 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

18. (original) The method of claim 16 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

19. (previously presented) The method of claim 14 wherein at least one of said discovered functional characteristics includes said software pipelined loop.

20. (original) The method of claim 19 wherein the allocation instruction inserted for said software pipelined loop is inserted above said loop in said control flow graph.

21. (previously presented) The method of claim 20 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

22. (original) The method of claim 21 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

23. (previously presented) The method of claim 14 wherein said one or more functional characteristics include a function call.

24. (previously presented) The method of claim 14 further comprising determining the number of on-processor_registers to be allocated for an allocation instruction after a functional characteristic is found.

25. (previously presented) The method of claim 24 wherein all functional characteristics from said group within said routine are discovered before said determining is performed.

26. (original) The method of claim 24 wherein said determining is performed before a next functional characteristic is discovered.

27. (original) The method of claim 14 further comprising building an understanding of said routine's control flow graph before said searching is performed.

28. (previously presented) A method, comprising:

performing a first allocation for a first amount of on-processor register space at the entry block of a routine, said first amount of on-processor register space for the use of said routine;

performing a second allocation for a second amount of on-processor register space for the storage of live information of said routine when said routine performs a function call to a second routine, said second amount less than said first amount;

performing said function call to said second routine; and,

performing a third allocation for a third amount of on-processor register space at the entry block of said second routine, said third amount of allocated on-processor register space for the use of said second routine and including a region of said first amount of allocated on-processor register space that stores stale information of said routine when said routine said performs said function call to said second routine.

29. (original) The method of claim 28 wherein said live information is determined by identifying information that is referred to before and after said function call.

30. (original) The method of claim 29 wherein said information identified after said function call extends to an exit block of said routine.

31. (original) The method of claim 30 wherein the worst case path to said exit block is allocated for.

32. (original) The method of claim 29 wherein said information identified after said function call extends to a post-dominator block of said routine.

33. (original) The method of claim 32 wherein the worst case path to said post-dominator block is allocated for.

34. (original) The method of claim 28 wherein said live information is information that is local to said routine.

35. (previously presented) The method of claim 34 wherein a processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and register space used only for global information, said allocation instruction pertaining only to said register space used for local information.

36. (original) The method of claim 28 wherein said live information includes global information.

37. (original) The method of claim 28 wherein said second allocation is performed just before said function call.

38. (original) The method of claim 28 wherein said second allocation is performed in a pre-dominator basic block of said function call.

39. (previously presented) The method of claim 38 wherein said second allocation is performed in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

40. (original) The method of claim 28 further comprising compiling said routine.

41. (currently amended) A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method comprising:

inserting an on-processor register allocation instruction within a machine code ~~version~~ description of a routine if a function call instruction to perform a function call to another routine is found within said routine, said inserted on processor register allocation instruction to allocate less on-processor register space for the use of said routine than an amount of on-processor register space allocated for the use of said routine by another on-processor register allocation instruction that is executed prior to said inserted on processor register allocation instruction.

42. (original) The machine readable medium of claim 41 further comprising instructions which cause a processor that executes said routine to configure said allocation instruction to allocate only for the live information that exists within said routine when said inserted allocation instruction is executed.

43. (original) The machine readable medium of claim 42 wherein said live information is determined by identifying information that is referred to before and after said function call.

44. (original) The machine readable medium of claim 43 wherein said information identified after said function call extends to an exit block of said routine.

45. (original) The machine readable medium of claim 44 wherein the worst case path to said exit block is allocated for.

46. (original) The machine readable medium of claim 43 wherein said information identified after said function call extends to a post-dominator block of said routine.

47. (original) The machine readable medium of claim 46 wherein the worst case path to said post-dominator block is allocated for.

48. (original) The machine readable medium of claim 42 wherein said live information is information that is local to said routine.

49. (previously presented) The machine readable medium of claim 48 wherein said processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and register space used only for global information, said allocation instruction pertaining only to said register space used for local information.

50. (original) The machine readable medium of claim 42 wherein said live information includes global information.

51. (previously presented) The machine readable medium of claim 41 wherein said allocation instruction is inserted just before a machine code representation of said function call.

52. (original) The machine readable medium of claim 41 wherein said allocation instruction is inserted in a pre-dominator basic block of said function call.

53. (original) The machine readable medium of claim 52 wherein said allocation instruction is inserted in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

54. (currently amended) A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method comprising:

inserting an on-processor register allocation instruction within a machine code description of a routine because a functional characteristic selected from the group consisting of:

a) a loop that exists within a control flow graph of said routine;

b) a software pipelined loop; and,

c) a function call to another routine;

is discovered within said routine, said inserted on-processor register allocation instruction to allocate less on processor register space for the use of said routine than had been previously allocated for the use of said routine.

55. (previously presented) The machine readable medium of claim 54 wherein at least one of said discovered functional characteristics includes said loop that exists within a control flow graph of said routine.

56. (previously presented) The machine readable medium of claim 55 wherein the allocation instruction inserted for said loop is inserted above said loop in said control flow graph.

57. (original) The machine readable medium of claim 56 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

58. (original) The machine readable medium of claim 56 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

59. (previously presented) The machine readable medium of claim 54 wherein at least one of said discovered functional characteristics includes said software pipelined loop.

60. (previously presented) The machine readable medium of claim 59 wherein the allocation instruction inserted for said software pipelined loop is inserted above said loop in said control flow graph.

61. (original) The machine readable medium of claim 60 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

62. (original) The machine readable medium of claim 61 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

63. (previously presented) The machine readable medium of claim 54 wherein said one or more functional characteristics include a function call.

64. (previously presented) The machine readable medium of claim 54 further comprising sequences of instructions which cause the system to determine the number of on-processor registers to be allocated for an allocation instruction after a functional characteristic is found.

65. (previously presented) The machine readable medium of claim 64 wherein functional characteristics from said group within said routine are discovered before said determining is performed.

66. (original) The machine readable medium of claim 64 wherein said determining is performed before a next functional characteristic is discovered.

67. (original) The machine readable medium of claim 54 further comprising sequences of instructions which cause the system to build an understanding of said routine's control flow graph before said searching is performed.

68. (previously presented) A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method, comprising:

performing a first allocation for a first amount of on-processor register space at the entry block of a routine, said first amount of on-processor register space for the use of said routine;

performing a second allocation for a second amount of on-processor register space for the storage of live information of said routine when said routine performs a function call to a second routine, said second amount less than said first amount;

performing said function call to said second routine; and,

performing a third allocation for a third amount of on-processor register space at the entry block of said second routine, said third amount of allocated on-processor register space for the use of said second routine and including a region of said first amount of allocated on-processor register space that stores

stale information of said routine when said routine said performs said function call to said second routine.

69. (previously presented) The machine readable medium of claim 68 wherein said live information is determined by identifying information that is referred to before and after said function call.

70. (previously presented) The machine readable medium of claim 69 wherein said information identified after said function call extends to an exit block of said routine.

71. (previously presented) The machine readable medium of claim 70 wherein the worst case path to said exit block is allocated for.

72. (previously presented) The machine readable medium of claim 69 wherein said information identified after said function call extends to a post-dominator block of said routine.

73. (previously presented) The machine readable medium of claim 72 wherein the worst case path to said post-dominator block is allocated for.

74. (previously presented) The machine readable medium of claim 68 wherein said live information is information that is local to said routine.

75. (previously presented) The machine readable medium of claim 74 wherein a processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and

register space used only for global information, said allocation instruction pertaining only to said register space used for local information.

76. (previously presented) The machine readable medium of claim 68 wherein said live information includes global information.

77. (previously presented) The machine readable medium of claim 68 wherein said second allocation is performed just before said function call.

78. (previously presented) The machine readable medium of claim 68 wherein said second allocation is performed in a pre-dominator basic block of said function call.

79. (previously presented) The machine readable medium of claim 68 wherein said second allocation is performed in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

80. (previously presented) The machine readable medium of claim 68 further comprising compiling said routine.

COMMENTS

The enclosed is responsive to the Examiner's Office Action mailed on May 6, 2004. At the time the Examiner mailed the Office Action claims 1 through 80 were pending. By way of the present response the Applicant has amended claims 1, 14, 41 and 54. As such claims 1 through 80 remain pending. The Applicant respectfully requests reconsideration of the present application and the allowance of claims 1 through 80.

102 Rejections

Independent Claims 28 and 68

Independent claims 28 and 68 stand rejected under 35 USC 102(b) as being anticipated by U.S. Patent No. 6,230,317 (hereinafter, "Wu"). In rejecting independent claim 28, the Examiner cites various aspects of Wu's teachings from Col. 3, line 64 through Col. 4, line 28. Independent claims 28 and 68 recite (emphasis added):

> 28. A method, comprising:
> performing a first allocation for a first amount of on-processor register space at the entry block of a routine, said first amount of on-processor register space for the use of said routine;
> performing a second allocation for a second amount of on-processor register space for the storage of live information of said routine when said routine performs a function call to a second routine, said second amount less than said first amount;
> performing said function call to said second routine; and,
> performing a third allocation for a third amount of on-processor register space at the entry block of said second routine, said third amount of allocated on-processor register space for the use of said second routine and including a region of said first amount of allocated on-processor register space that stores stale information of said routine when said routine said performs said function call to said second routine.

68. A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method, comprising:
performing a first allocation for a first amount of on-processor register space at the entry block of a routine, said first amount of on-processor register space for the use of said routine;
performing a second allocation for a second amount of on-processor register space for the storage of live information of said routine when said routine performs a function call to a second routine, said second amount less than said first amount;
performing said function call to said second routine; and,
performing a third allocation for a third amount of on-processor register space at the entry block of said second routine, said third amount of allocated on-processor register space for the use of said second routine and including a region of said first amount of allocated on-processor register space that stores stale information of said routine when said routine said performs said function call to said second routine.
.

"To anticipate a claim, the reference must teach every element of the claim" MPEP 2131. Referring to the emphasized claim language of Claims 28 and 68, it is clear that claims 28 and 68 include the following elements:

1) first and second register space allocations for a routine;

2) less register space being allocated for the routine by the second register allocation as compared to the first.

Wu fails to disclose any such matter.

Firstly, Wu only teaches a single allocation instruction per routine and not "first" and "second" allocation instructions for a particular routine. Specifically, referring to Figure 3 of Wu, register space 310 is allocated for procedure/routine A, register space 320 is allocated for procedure/routine B and register space 330 is allocated for procedure C. See, Wu Col. 4, lines 9-11. Importantly, Wu clearly states that register space is only allocated once for a procedure (i.e., not twice) as a consequence of that procedure being called by another procedure. See, Wu Col. 4, lines 11-13 ("[t]he register windows 320 and 330 are formed by

procedure A calling procedure B and procedure B calling procedure C"). It is therefore clear from the figures and the express statements of Wu that first and second allocations are not made for any particular routine. Therefore the first of the above cited claim elements of independent claims 28 and 68, first and second register allocations for a routine, is clearly not disclosed by Wu.

Secondly, although Wu teaches that the register space for two different procedures/routines can be shared for purposes of passing parameters between them (See, Wu Col. 4, lines 17-21), Wu's sharing of register space does not amount to the allocation of "less" register space for a particular routine. For example, again referring to Figure 3 of Wu, register space 310 remains fixed in its allocation for procedure/routine A. Because there is no lessening of allocated register space for procedure/routine A, and because the teachings of Wu are analogous in this regard with respect to procedures/routines B and C, Wu fails to disclose the lessening of allocated register space for a particular routine. Therefore, the second of the above cited claim elements of independent claims 28 and 68, lessening of allocated register space for a routine, is clearly not disclosed by Wu.

Therefore, in least two instances, the portion of Wu cited by the Examiner fails to anticipate the subject matter claimed by Applicant.

Moreover, the Examiner has responded to the Applicant's argument is support of claims 28 and 68 filed on 2/19/04 in response to the Examiner's Office Action mailed 11/20/03. In the Office Action mailed 5/6/03, the Examiner

repeated the following portion of the Applicant's 2/19/04 response (emphasis added):

> The Applicant respectfully submits that independent claim 28 is patentable over Wu for the same reasons discussed above with respect to independent claim 28[sic]. That is, claim 28 recites a pair of allocations ("a first allocation" and "a second allocation") that allocate registers for the use of the same routine. Moreover, claim 28 recites that the later (second) allocation allocates for less register space than the earlier allocation.
>
> These claimed features stem at least from the Applicants' discussion of Figures 5 and 6 of the present application. That is, Figures 5 and 6 of the present application and their surrounding discussion disclose a technique where an allocation instruction that allocates register space (e.g., space 610 in Figure 6) for a caller routine (as a consequence of the caller routine having a function call) can be configured to allocate for less register space than a previously executed allocation instruction allocated (e.g., space 605 of Figure 6) for the same caller routine (e.g., located at the caller routine's entry block).
>
> By contrast the Wu reference at best only discloses, teaches or suggests an allocation instruction for the called routine rather than the caller routine. As such, the Wu reference fails to cover the Applicants' claimed subject matter. The Wu reference teaches the implementation of a software pipelined loop as a called sub function and the corresponding allocation of register space for the called sub function. See, Wu Col. 3, lines 26-29 and Col.4 lines 5-21. Nothing is said in Wu with respect to the insertion of an allocation instruction that allocates register space for the caller function, however. Therefore the Wu reference fails to anticipate each and every limitation of independent claim 28 of the present application.
>
> Therefore independent claim 28 and its corresponding dependent claims are patentable over the Wu reference.

The Examiner provided the following response:

> As in the argument for claim 1, the Applicant distinguishes between a caller routine and a called routine, but does not make that distinction in the claims. If this distinguishes the invention over the prior art, then that feature should be in the claims.

See, Office Action mailed 5/6/03 pgs. 17-18.

The Applicant respectfully submits that although the routine of claims 28 and 68 can be construed as a caller routine because it explicitly recites the calling of a second routine, the Applicant's **leading defense of claims 28 and 68 emphasized above (that claim 28 recites first and second allocations for**

register space for a particular routine; and, recites that the second allocation allocates less register space than the first) identifies an even more basic distinction between the teachings of Wu and the subject matter claimed by claims 28 and 68. The Examiner appears to have completely avoided this distinction on the merits by focusing instead on the Applicant's reference to a caller routine. The Applicant respectfully requests the Examiner to provide a meaningful analysis, provide a new theory of rejection; or, allow independent claims 28 and 68.

Independent Claim 1

According to the Examiner independent claim 1 has been rejected "for the same reasons put forth in the rejection of claim 37". See, Office Action mailed 5/6/04, pg. 5. Claim 37 recites:

> 37. The method of claim 28 wherein said second allocation is performed just before said function call.

In rejecting claim 37 the Applicant stated "[t]he second storage area is allocated before the function is called (column 4, lines 9-14)". See, Office Action mailed 5/6/04, pg. 5. Independent Claim 1 presently recites (emphasis added):

> 1. A method, comprising:
> inserting an on-processor register allocation instruction within a machine code description of a routine if a function call instruction to perform a function call to another routine is found within said routine, said inserted on processor register allocation instruction to allocate less on-processor register space for the use of said routine than an amount of on-processor register space allocated for the use of said routine by another on-processor register allocation instruction that is executed prior to said inserted on processor register allocation instruction.

From the emphasized claim language above it is clear that the same pair of claim elements discussed above with respect to claims 28 and 68 also exist with respect to claim 1. That is claim 1 (and 41) include the following elements:

1) first and second register space allocations for a routine;

2) less register space being allocated for the routine by the second register allocation as compared to the first.

Note that these claim elements existed prior to the amendments made to claims 1 and 41 by way of the present office action response. As a consequence, these amendments are not substantially related to the patentability of these claims.

Because the theory of rejection applied against claim 1 by the Examiner incorporates the same theory applied against claim 28 (by way of claim 37's dependence on claim 28), and because claim1 effectively includes the same above pair of elements included by claim 28 and discussed at length above, the Examiner has failed to articulate a valid theory of rejection against claim 1. That is, the Wu reference not only fails to disclose first and second register allocations for a routine but also fails to disclose the lessening of allocated register space for a routine.

Moreover, the Examiner has responded to the Applicant's argument is support of claim 1 filed on 2/19/04 in response to the Examiner's Office Action mailed 11/20/03. In the Office Action mailed 5/6/04, the Examiner repeated the following portion of the Applicant's 2/19/04 response (emphasis added):

> Claim 1 recites a pair of allocation instructions ("an on-processor register allocation instruction" and "another on-processor register allocation instruction") that allocate registers for the use of the same routine. Moreover,

claim 1 recites that the later executed allocation instruction allocates for less register space than the earlier executed allocated allocation instruction.

These claimed features stem at least from the Applicants' discussion of Figures 5 and 6 of the present application. That is, Figures 5 and 6 of the present application and their surrounding discussion disclose a technique where an allocation instruction that allocates register space (e.g., space 610 in Figure 6) for a caller routine (as a consequence of the caller routine having a function call) can be configured to allocate for less register space than a previously executed allocation instruction allocated (e.g., space 605 of Figure 6) for the same caller routine (e.g., located at the caller routine's entry block).

By contrast the Wu reference at best only discloses, teaches or suggests an allocation instruction for the called routine rather than the caller routine. As such, the Wu reference fails to cover the Applicants' claimed subject matter. The Wu reference teaches the implementation of a software pipelined loop as a called sub function and the corresponding allocation of register space for the called sub function. See, Wu Col. 3, lines 26-29 and Col.4 lines 5-21. Nothing is said in Wu with respect to the insertion of an allocation instruction that allocates register space for the caller function, however. Therefore the Wu reference fails to anticipate each and every limitation of independent claim 1 of the present application.

The Examiner provided the following response:

Firstly claim 1 refers to inserting an on-processor register allocation instruction, and the applied art reads on that action. There are no other steps in the claim other than inserting. Simply appending more information onto the claim does not necessarily add patentable weight to the single step of "inserting an on-processor register allocation instruction." A question arises: Are there any other analyses or steps to determine how much space is less than previously allocated? It is not clear in the claim how this gets down.

Also, the Applicant makes a distinction between a caller routine and a called routine, which is not even mentioned in the claim; therefore it appears to be a moot argument. If this distinguished the invention over the prior art, then that feature should be in the claims.

See Examiner's Office Action mailed 5/6/04, pgs. 14-16.

The Applicant respectfully requests the Examiner to cite the legal authority (e.g., MPEP section, case law citation, etc.) that supports the Examiner's contention that "[s]imply appending more information onto the claim does not necessarily add patentable weight". The Applicant does not believe there exists any such legal authority and, on the belief that the Applicant

is correct, the Applicant responds to the Examiner by asserting that the Examiner is responsible for finding in the prior art every element of the Applicant's claim "To anticipate a claim, the reference must teach every element of the claim" MPEP 2131.

Here, in order to avoid a substantive comparison of the teachings of the prior art and the matter being claimed, the Examiner appears to have "conjured up" a bizarre legal theory that only language relating to verbs that end in "ing" need to be considered by the Examiner. The Examiner is not permitted to make up the law so that a substantive discussion on the merits can be conveniently avoided. The Applicant respectfully submits that all elements in the claim must be considered and dealt with and requests the Examiner to do so.

With respect to the Examiner's question regarding "any other analyses or steps to determine how much space is less than previously allocated", the Applicant responds that claim 1 as written meets the requirements of 35 USC 112 paragraphs 1 and 2. Therefore: 1) the Applicant had possession of the matter claimed by claim 1 at the time the Applicant's specification was filed as evidenced by the Applicant's specification itself; 2) claim 1 is enabled by the Applicant's specification; 3) the best mode for implementing the subject matter claimed by claim 1 at the time the Applicant filed the present application is disclosed in the Applicant's specification; and, 4) claim 1 particularly points out and distinctly claims the subject matter regarded as the invention to which claim 1 pertains.

The Examiner has only provided a rejection under 35 USC 102 which merely deals with a comparison between the teachings of the prior art and the subject matter claimed by claim 1. The Examiner's question does not invoke any analysis of the prior art and therefore stands outside the relevant realm of the Examiner's own rejection. Therefore the Applicant need not address the Examiner's question.

Moreover, the Applicant respectfully submits that although the routine of claim 1 can be construed as a caller routine because it explicitly recites the calling of a another routine, the Applicant's **leading defense of claim 1 in the above cited portion of the Applicant's 2/19/04 response emphasized above (that claim 1 recites a pair of allocation instructions that allocate registers for the use of the same routine and recites that the later executed allocation instruction allocates for less register space than the earlier executed allocated allocation instruction) identifies an even more basic distinction between the teachings of Wu and the subject matter claimed by claim 1**. As with claim 28, the Examiner appears to have completely avoided this distinction on the merits by focusing instead on the Applicant's reference to a caller routine. The Applicant respectfully requests the Examiner to provide a meaningful analysis, provide a new theory of rejection; or, allow independent claim 1.

103 Rejections

Independent Claims 1 and 41

Independent claims 1 and 41 have been rejected under 35 USC 103 as being unpatentable over the combination of U.S. Patent No. 5,367,684 (hereinafter, "Smith") in view of Wu. Claims 1 and 41 recite:

> 1. A method, comprising:
> inserting an on-processor register allocation instruction within a machine code description of a routine if a function call instruction to perform a function call to another routine is found within said routine, said inserted on processor register allocation instruction to allocate less on-processor register space for the use of said routine than an amount of on-processor register space allocated for the use of said routine by another on-processor register allocation instruction that is executed prior to said inserted on processor register allocation instruction.

> 41. A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method comprising:
> inserting an on-processor register allocation instruction within a machine code description of a routine if a function call instruction to perform a function call to another routine is found within said routine, said inserted on processor register allocation instruction to allocate less on-processor register space for the use of said routine than an amount of on-processor register space allocated for the use of said routine by another on-processor register allocation instruction that is executed prior to said inserted on processor register allocation instruction.

As discussed at length above, claims 1 and 41 include the following claim elements:

1) first and second register space allocations for a routine;

2) less register space being allocated for the routine by the second register allocation as compared to the first.

The Wu reference, also discussed at length above, fails to cover either of these claim elements. The Applicant believes Smith is incapable of disclosing,

teaching or suggesting at least the second of the above pair of elements (i.e., the lessening of a routine's register space). Here, the Applicant's original characterization of Smith that was presented in the Applicant's Office Action response filed on 2/19/04 is somewhat usefull.

> Smith teaches a register allocation method (specifically, the allocation of register space to two "register candidates" that are live within the same basic block of instructions through an improved "register candidate usage matrix". See, Smith, Col. 2, lines 5-32; Col.2, line 60 – Col. 3, line 2) but says nothing about the insertion of an allocation instruction. That is, Smith teaches a method that can perhaps be used to determine "how many" registers are to be allocated for by an allocation instruction; but, by contrast, fails to disclose how many allocation instructions are to be inserted into a routine and/or a function call made from the routine being a stimulus for the insertion of an allocation instruction into the routine.
>
> See, Applicant's Office Action response 2/19/04, pg. 18.

Moreover, beyond Smith's silence with respect to actual allocation instructions, there appears to be no discussion in Smith concerning the lessening of register space allocated for a routine. That is, the Applicant is unable to find any disclosure in Smith that shows the reduction of a routine's previously allocated register space. The Applicant respectfully requests the Examiner to point out to the Applicant where such disclosure in Smith is found. If such a disclosure cannot be found, claim 1 is patentable over the Examiner's combination.

Independent Claims 15 and 54

Independent claims 15 and 54 stand rejected under 35 USC 103 as being unpatentable over the combination of the We and Smith references. Independent claims 15 and 54 recite (emphasis added).

14. A method comprising:
inserting an on-processor register allocation instruction within a machine code description of a routine because a functional characteristic selected from the group consisting of:
a) a loop that exists within a control flow graph of said routine;
b) a software pipelined loop; and,
c) a function call to another routine;
is discovered within said routine, said inserted on-processor register allocation instruction to allocate less on processor register space for the use of said routine than had been previously allocated for the use of said routine.

54. A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method comprising:
inserting an on-processor register allocation instruction within a machine code description of a routine because a functional characteristic selected from the group consisting of:
a) a loop that exists within a control flow graph of said routine;
b) a software pipelined loop; and,
c) a function call to another routine;
is discovered within said routine, said inserted on-processor register allocation instruction to allocate less on processor register space for the use of said routine than had been previously allocated for the use of said routine.

Each of claims 14 and 54 recite the lessening of allocated register space for the use of a routine. The failure of both the Wu and Smith references to disclose, teach or suggest the lessening of allocated register space for the use of a routine have been discussed in detail above.

Specifically, although Wu teaches that the register space for two different procedures/routines can be shared for purposes of passing parameters between them (See, Wu Col. 4, lines 17-21), Wu's sharing of register space does not amount to the allocation of "less" register space for a particular routine. For example, again referring to Figure 3 of Wu, register space 310 remains fixed in its allocation for procedure/routine A. Because there is no lessening of allocated register space for procedure/routine A, and because the teachings of Wu are analogous in this regard with respect to procedures/routines B and C, Wu fails to

disclose the lessening of allocated register space for a particular routine. Therefore, the lessening of allocated register space for a routine, is clearly not disclosed by Wu.

There also appears to be no discussion in Smith concerning the lessening of register space allocated for a routine. The Applicant is unable to find any disclosure in Smith that shows the reduction of a routine's previously allocated register space.

As such, the Applicant respectfully submits that claims 14 and 54 are patentable over the combination of Wu and Smith.

Closing Comments

The Applicant has demonstrated that all independent claims are patentable over the rejections that have been applied against them. Therefore all claims are allowable and the Applicant respectfully requests the allowance of same.

The Applicant's silence to the dependent claims should not be construed as an admission by the Applicant that the Applicant is complicit with the Examiner's rejection of these claims. Because the Applicant has demonstrated the patentability of the independent claims, the Applicant need not substantively address the theories of rejection applied to the dependent claims.

CONCLUSION

In light of the foregoing comments the Applicants respectfully submit that claims 1 through 80 are patentable and the Applicants respectfully request the allowance of same.

Applicants respectfully submit the present application is in condition for allowance. If the Examiner believes a telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call Robert O'Rourke at (408) 720-8300.

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: 9/2/04

Thomas C. Webster
Reg. No. 46,154

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300